**Practical - 3**

**Aim:**  Control High Voltage Electrical Devices Using Esp8266 NodeMCU.

Connect few electrical devices to NodeMCU and control the devices.

**Components:** LED, Relay Switch, Jumper wires, ESP 8266 NodeMCU

**Procedure:**

* Connections for ESP8266:

Relay -Ve to GND

Relay +Ve to 3v3

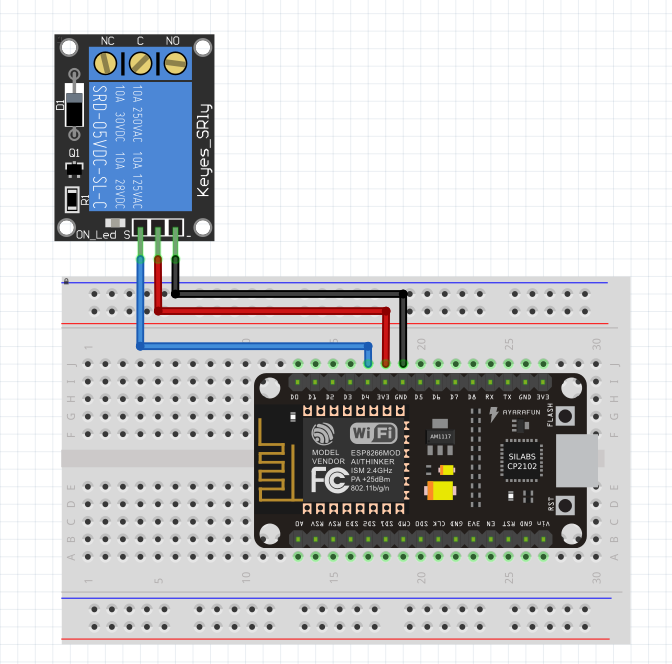
Relay input to D4 (pin 2)

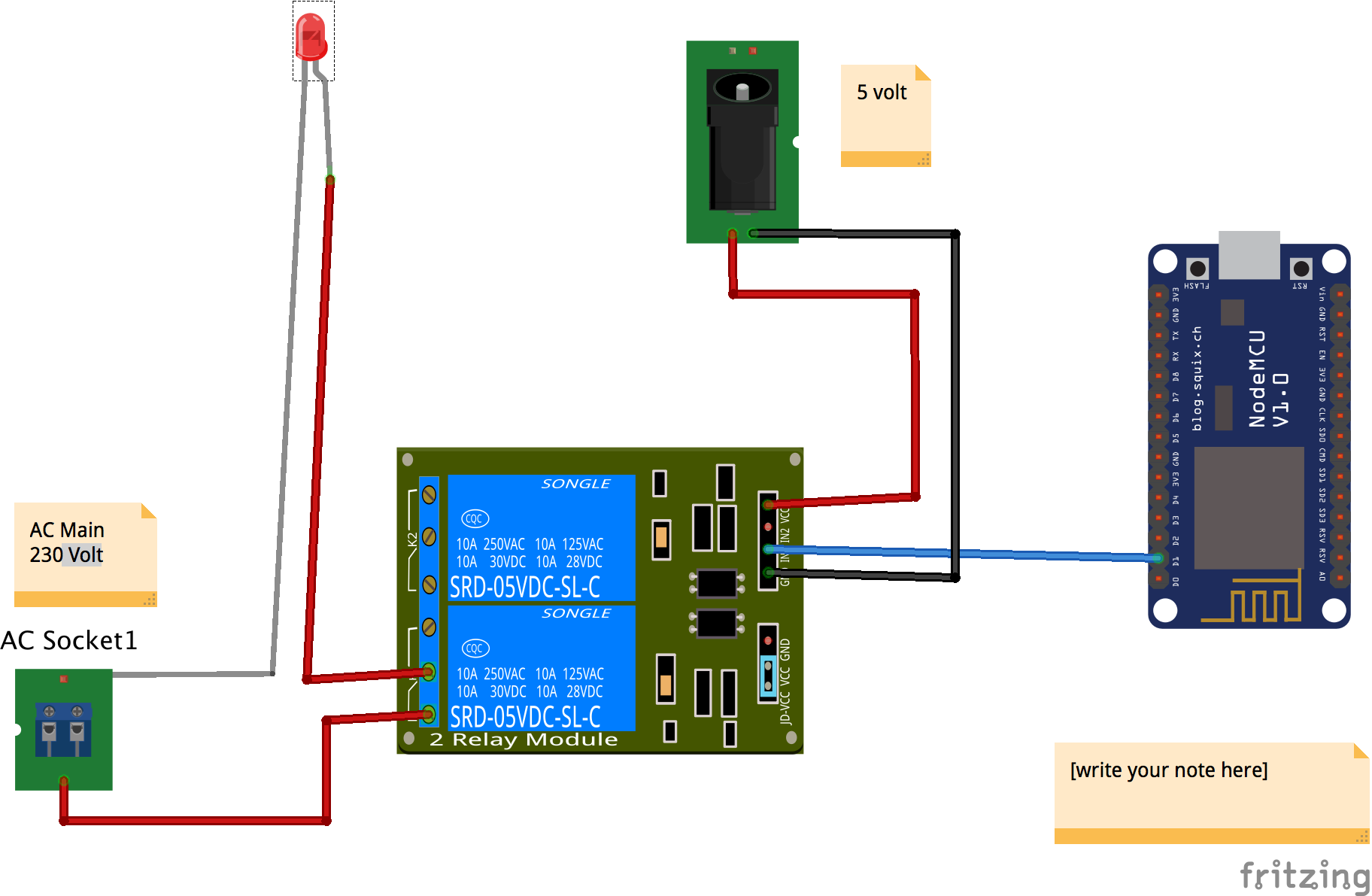
* Connections for Relay Switch:

LED’s -Ve to C (Comman)

LED’s +Ve to NC ( Normal Close)

* Implementation in diagram





* Code:

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

const char\* ssid = "NU"; // Replace with your network credentials

const char\* password = "12345678";

ESP8266WebServer server(80); //instantiate server at port 80 (http port)

String page = "";

int LEDPin = 13;

void setup(void){

page = "<h1>Simple NodeMCU Web Server</h1>

<p><a href=\"LEDOn\"><button>ON</button></a>&nbsp;

<a href=\"LEDOff\"><button>OFF</button></a>

</p>"; // the HTML of the web page

//make the LED pin output and initially turned off

pinMode(D7, OUTPUT);

digitalWrite(D7, LOW);

delay(1000);

Serial.begin(115200);

WiFi.begin(ssid, password); //begin WiFi connection

Serial.println("");

// Wait for connection

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("Connected to ");

Serial.println(ssid);

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

server.on("/", [](){

server.send(200, "text/html", page);

});

// the href tag on “On” button transfer the control to this function which make the output HIGH

server.on("/LEDOn", [](){

server.send(200, "text/html", page);

digitalWrite(13, HIGH);

delay(1000);

});

//the href tag on “Off” button transfer the control to this function which make the output LOW

server.on("/LEDOff", [](){

server.send(200, "text/html", page);

digitalWrite(13, LOW);

delay(1000);

});

server.begin();

Serial.println("Web server started!");

}

void loop(void){

server.handleClient();

}